Technical Note: Random Projection Direction Quantization on Hyperspheres

Overview

Most random projection methods retain the projected coordinates or threshold them independently (for example, random hyperplane locality-sensitive hashing).

An alternative viewpoint is to regard random projections as defining a **low-dimensional observation of direction** rather than a low-dimensional representation of the original vector.

The key idea is:

- Normalize the input vector.
- Compute a small number of random projections.
- Discard the projection magnitude.
- Quantize only the **direction** within the projected subspace.

This naturally leads to partitions of the original hypersphere.

Two Random Projections

Given a normalized input vector

$$v \in S^{d-1},$$

choose two random projection vectors

$$a,\ b.$$

Compute

$$x = a^T v, \qquad y = b^T v.$$

Instead of using the values directly, compute

$$\theta = \operatorname{atan2}(y, x).$$

Divide the circle into n equal angular sectors.

The bucket is simply

$$\text{bucket} = \left\lfloor \frac{n\,\theta}{2\pi} \right\rfloor.$$

Magnitude is Removed

Since

$$\operatorname{atan2}(\alpha y, \alpha x) = \operatorname{atan2}(y, x),$$

only direction remains.

The construction therefore operates on the surface of the unit hypersphere rather than throughout Euclidean space.

Geometric Interpretation

Rather than thinking about projections first, it is often easier to think geometrically.

1. Select a random two-dimensional plane through the origin.
2. The plane intersects the hypersphere in a great circle.
3. Divide the great circle into equal angular sectors.
4. Extend each sector into a cone.
5. Intersect the cone with the hypersphere.

Each cone defines one bucket.

The projection computation is simply an efficient way of determining which cone contains the input vector.

Bucket Boundaries

The boundaries are hyperplanes through the origin.

Intersecting these hyperplanes with the hypersphere produces **great hyperspheres**.

Examples:

- 3D sphere → boundaries are great circles.
- 4D sphere → boundaries are ordinary 2-spheres.
- Higher dimensions follow the same pattern.

Each bucket is the region of the hypersphere lying between two neighboring boundaries.

Three Random Projections

Instead of two projections, choose three:

$$x = a^T v, \qquad y = b^T v, \qquad z = c^T v.$$

Normalize

$$(x, y, z) \to \frac{(x, y, z)}{\sqrt{x^2 + y^2 + z^2}}.$$

The normalized point lies on an ordinary sphere.

The question becomes:

> Where on the surface of this random sphere does the input vector map?

Instead of dividing a circle into angular sectors, we now partition the sphere.

Possible spherical partitions include:

- latitude/longitude grids
- equal-area grids
- geodesic sphere subdivisions
- spherical Voronoi diagrams
- HEALPix
- Fibonacci sphere partitions

Any partition of the sphere becomes a partition of the original hypersphere.

Generalization

Using k random projections,

$$S^{d-1} \to S^{k-1}.$$

Normalize the projected vector.

The problem becomes:

> Which cell of a partition of S^{k-1} contains the projected point?

This forms a family of direction quantizers.

- $k = 2$: circle partitions.
- $k = 3$: sphere partitions.
- Larger k: partitions of higher-dimensional spheres.

Why Random Projections Work Well

In high dimensions, independently chosen random vectors are almost orthogonal.

Typical inner products are approximately

$$O\left(\frac{1}{\sqrt{d}}\right).$$

Consequences:

- projection coordinates contain little redundant information
- each projection measures a nearly independent direction
- the projected coordinate system is close to orthogonal
- no direction is preferred statistically

This greatly simplifies the interpretation of the projected sphere.

Relationship to Existing Methods

Traditional random hyperplane hashing asks a binary question:

> Which side of a hyperplane is the vector on?

The atan2 construction asks:

> What angular direction does the projected vector have?

This naturally produces multi-way buckets instead of binary decisions.

Possible Advantages

Potential advantages include:

- richer directional information per projection group
- controllable bucket count
- rotational symmetry
- natural operation on normalized vectors
- easy implementation
- compatibility with random projection methods
- potentially useful context generation for Atlas neural networks and related architectures

---

Mathematical Areas Related to This Idea

Readers interested in the underlying mathematics may wish to look up:

- hyperspheres
- great circles
- great hyperspheres

- Grassmann manifolds
- random subspaces
- directional statistics
- von Mises-Fisher distribution
- spherical geometry
- spherical tessellations
- spherical Voronoi diagrams
- hyperplane arrangements
- integral geometry
- stochastic geometry
- locality-sensitive hashing
- Johnson-Lindenstrauss embeddings
- random matrix theory

Research Questions

Some questions that appear worth investigating include:

- How should the projected circle or sphere be partitioned for different objectives?
- What partition shapes maximize locality preservation?
- Are equal-area partitions preferable to equal-angle partitions?
- How does collision probability compare with binary hyperplane LSH?
- How many projection dimensions k provide the best trade-off between discrimination and computational cost?
- What is the relationship between bucket entropy and partition geometry?
- Can multiple independent projected spheres be combined efficiently?
- How uniformly do real-world datasets occupy the projected sphere?
- Can bucket occupancy statistics guide adaptive partition refinement?
- Can spherical partitions be learned from data instead of fixed in advance?
- How do these constructions interact with decision-tree or context-based neural network architectures?
- Are there theoretical guarantees analogous to those available for classical random projection methods?

Summary

Random projections are usually viewed as dimensionality reduction.

An alternative viewpoint is that a small set of random projections defines a randomly oriented low-dimensional sphere on which the input direction can be located.

Partitioning this projected sphere induces a partition of the original hypersphere.

This provides a simple geometric framework that generalizes the atan2 construction from circle partitions to arbitrary spherical partitions while remaining computationally inexpensive and naturally suited to normalized high-dimensional vectors.